This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2007 proceedings.

Measurement-Based Self Organization of Interfering 802.11 Wireless Access Networks

Bruno Kauffmann, François Baccelli, Augustin Chaintreau,
Vivek Mhatre, Konstantina Papagiannaki, Christophe Diot

Abstract— **The popularity of IEEE 802.11 WLANs has led to dense deployments in urban areas. High density leads to sub-optimal performance unless the interfering networks learn how to optimally use and share the spectrum. This paper proposes two fully distributed algorithms that allow (i) multiple interfering 802.11 Access Points to select their operating frequency in order to minimize interference, and (ii) users to choose the Access Point they attach to, in order to get their fair share of the whole network bandwidth. The proposed algorithms rely on Gibbs sampler, and do not require explicit coordination among the wireless devices. They only require the participating wireless nodes to measure local quantities such as interference and transmission delay. The algorithms are shown to lead to optimal bandwidth sharing, where optimality is defined according to the minimal potential delay. We analytically prove the convergence of the proposed algorithms, and study their performance by simulation.**

I. INTRODUCTION

The increasing popularity of Wireless Local Area Networks (WLANs) has led to a dramatic increase in the density of WiFi Access Points (APs) in university campuses, enterprise environments, public places and homes. High node density results in increased interference, and overall sub-optimal user throughput due to contention [1]. In many instances of such environments, clients find themselves within range of a number of APs that may belong to the same administrative authority as the user, offer service for a charge, or combinations of the above. The selection of the operating frequency by APs, and the association of users to APs dictates the overall network capacity and user performance due to the shared nature of the 802.11 medium. In this work, we advocate that WiFi networks need to become *self-organizing* to make optimal use of the shared spectrum. The above is not possible in today's WiFi networks because:

1) User devices are programmed to associate with the AP with the strongest received signal strength. This leads to scenarios where some APs have very few users, while other APs are overloaded with many users.
2) Multiple APs in close proximity of each other often use the same (or overlapping) channel due to the default out-of-the-box channel settings. This results in increased interference and poor user throughput (even with intelligent user association algorithms).

We show that significant improvement can be achieved in user throughput via: (i) interference mitigation through optimum *channel selection* algorithms, and (ii) optimal bandwidth sharing through fair *user association* algorithms.

The state of the art channel selection algorithms in managed WiFi networks use a *centralized* approach that collects information from the entire network, and derives the optimal configuration. Such an approach is not scalable due to the NP-hard nature of the problem, and requires a separate processing infrastructure for performing the centralized computations (see Aruba networks, Symbol, Cisco). In unmanaged WiFi networks, the channel selection algorithm consists of using the default factory setting, while the user association is based on strongest received signal strength.

We focus on unmanaged WiFi networks where (i) all the participating devices belong to the same enterprise, or (ii) there is an implied co-operative relationship (e.g. a community network, free hotspots, etc.), so that users can associate with any AP in the network. Our optimization criterion for channel selection is the minimization of global interference, and for user association is the *minimal potential delay* of the users (defined in [2]). The latter quantity captures the long-term throughput that a user should expect to receive from a fully saturated network. In the proposed algorithms, local decision procedures are driven by actual measurements thereby avoiding simplifying assumptions about the nature of the wireless medium and the impact of the MAC protocol. Our algorithms do not require changes at the MAC layer. They can be implemented through simple software modifications, and can be supported by the wireless devices through firmware upgrades (in line with current proposals within the IEEE 802.11k task group). An actual implementation on a small scale testbed validates the feasibility of our scheme.

The rest of the paper is organized as follows. The next section discusses related work. In Section III, we formulate the problem, introduce notation and assumptions. In Section IV, we describe the proposed algorithms. Their performance is then extensively simulated and compared to currently used strategies in Section V. Results over a proof of concept testbed are presented in Section VI. We conclude and discuss future work in Section VII.

II. RELATED WORK

Centralized algorithms for channel selection and user association are proposed in [3] and [4] respectively. In a similar

B. Kauffmann is with ENS, Paris, France. F. Baccelli is with INRIA-ENS, Paris, France. Email: {bruno.kauffmann, francois.baccelli}@ens.fr. K. Papagiannaki is with Intel Research Pittsburgh, USA. Email: dina.papagiannaki@intel.com. A. Chaintreau, V. Mhatre, and C. Diot are with Thomson, Paris, France. Email: {augustin.chaintreau, vivek.mhatre, christophe.diot}@thomson.net.

spirit, some papers have addressed optimal channel allocation in wireless meshed networks [5]. In [6], the authors propose a distributed channel hopping scheme which results in improved fairness as compared to static channel allocation schemes. The problem of channel assignment in WLANs is also very briefly addressed in [1], but was found to have limited impact in terms of performance; APs are recommended to select an orthogonal frequency that none of their neighbors uses if possible. Kumar and Kumar [7] study the user association problem in 802.11 networks by casting it as a utility optimization problem. Korakis et al. [8] also address the problem of user association, and propose a greedy algorithm that locally maximizes the throughput of a user based on the estimation of its SINR value.

To the best of our knowledge, the present paper is the first to propose a set of algorithms that simultaneously solves the problems of *channel selection* and *user association* in a *fully distributed* way. These two problems have only very recently been addressed together by Mishra et al. (see [9]). However, the authors in [9] use a *centralized* approach. Some commercial products[1] claim that they have a solution to these problems, but the technology used is not disclosed. Our approach is unique because our algorithms are fully distributed, and are analytically shown to lead to a globally optimal bandwidth sharing.

III. PROBLEM SETUP

In this work we look into distributed algorithms for *optimal* channel selection and user association. We acknowledge that the best overall solution would jointly optimize the channel selection and user association. However, requiring the entire AP infrastructure to adapt to the small timescale over which the user population dynamics vary may lead to stability problems. Hence, we believe that from a system design perspective, AP frequency selection should not depend on user associations across the network, since it allows for a simpler implementation, and is more robust to fast user arrivals and departures.

A. Assumptions and Notation

We assume a network model where a user can associate with any AP in the network. The exact functionality required by APs and users to support our algorithms is discussed in Section VI. The wireless access network that we consider in this paper is described by : (i) a set of APs $a \in \mathcal{A}$, (ii) a set of users $u \in \mathcal{U}$, and (iii) a set of available channels $c \in \mathcal{C}$.

- Let $c_a \in \mathcal{C}$ be the channel that is chosen by access point $a \in \mathcal{A}$. We introduce the function $s_{\mathrm{CH}}(a, b)$ that is equal to 1 if a and b are operating on the same channel, and to 0 if they use non-overlapping channels.

We assume in this paper that APs select an operating frequency among their band's non-overlapping frequencies. Our work can be easily extended to the case of partially overlapping channels (see [10]).

- Let $a_u \in \mathcal{A}$ be the AP that is chosen by user $u \in \mathcal{U}$. For each AP a, we introduce $\mathcal{U}_a \subseteq \mathcal{U}$ the subset of users associated with AP a. Note that the collection of these subsets is a partition of the set $\mathcal{U}$. We denote by K_a the number of users associated with a. For all pairs of users u and v, let $s_{\mathrm{AP}}(u, v)$ be the function whose value is equal to 1 if u and v are associated with the same AP, and to 0 in all other cases.

[1] see e.g. http://www.propagatenet.com/.

The AP channel assignment problem is choosing a collection $(c_a)_{a\in\mathcal{A}}$ in $\mathcal{C}$ satisfying a given criterion. Similarly the user association problem is choosing a collection $(a_u)_{u\in\mathcal{U}}$ satisfying a given criterion. These two criteria define a performance objective that we present in Section IV.

B. A Model for Bandwidth Sharing in 802.11

We define a cell as an AP together with all users that are associated to this AP. For simplicity, we focus on the downlink, where data is sent by APs to users, that is the majority of wireless traffic today. Extending the analysis to incorporate the uplink traffic is part of future work. We assume that the wireless links form the bottleneck, and that the network is fully saturated, i.e., the APs always have data to send to all the users.

If multiple APs coexist in the same frequency and collision domain, medium access is coordinated following the specifications of the 802.11 MAC protocol. The actual long-term throughput of a user in a cell then depends on three factors: (i) the medium utilization, or channel access time, obtained by the AP; (ii) the number of users in the cell, which determines how frequently each user is sent a data unit, and (iii) the quality of the wireless link from the AP to the user, which determines the data rate at which the user is served by the AP [11].

We assume that information is transmitted to each user in data units of the same length, so that the data unit transmission delay (the delay experienced by a data unit sent from the AP to user) of user u is given by

$$d(u) = \frac{1}{f(\mathrm{SINR}(u))},$$

where $f(\mathrm{SINR}(u))$ gives the *instantaneous* transmission rate on the channel from a_u to u, that is expressed in data units per second. The *auto-rate function* of 802.11 adapts the data rate $f(.)$ of the transmitter to the user SINR.

If AP a has other APs in its contention domain, its channel access time $M(a)$ will not be 100%, and its actual capacity will only be the fraction $M(a)$ of the medium capacity. In such a setting the max-min fair allocation of bandwidth in the cell implies that the long-term throughput obtained by each user u associated with a is given by:

$$r_u = \frac{M(a)}{\sum_{v\in\mathcal{U}_a} d(v)} = \frac{M(a)}{\sum_{v\in\mathcal{U}} s_{\mathrm{AP}}(u, v) d(v)} \ . \tag{1}$$

Note that despite the fact that the time to transmit the same unit of information is different from one user to another in the same cell, all the users of the same cell receive the same long-term throughput [12]. The inverse of r_u (which is identical for all users associated with the same AP) will be referred to as the *aggregated transmission delay* of the AP in what follows.

IV. Network Self-Configuration using Gibbs Samplers

We now propose algorithms for channel selection and user association. The algorithms are based on a simulated annealing technique called the *annealed Gibbs sampler*. In the following sub-section, we describe the basic notions and the framework of a Gibbs sampler. Details are available in [13].

A. The Gibbs Sampler

Consider an undirected graph with K nodes. Two nodes are said to be neighbors if they are connected by an edge. Each node is endowed with a state variable that belongs to a finite set $\mathcal{S}$. The state of the graph is the vector $\mathbf{s} = (s_1, \ldots, s_K)$ of the states of its nodes. For example, for the channel selection algorithm, the nodes are the APs, edges connect APs within communication range and the state of a node is its channel.

An energy function $\mathcal{E}$ associates a real number $\mathcal{E}(\mathbf{s})$ to each state $\mathbf{s}$ of the graph. The objective is to find one of the states that minimizes this energy function. In general, this problem is of combinatorial complexity and difficult to solve for large networks. However, if the energy function has a certain form (explained below), then the minimization can be achieved by using an annealed Gibbs sampler. For this, we define the following terms. A clique of order k is a set of nodes of cardinality k such that all pairs of nodes in the set are neighbors. Let C_k denote the set of all cliques of order k. A potential V on this graph is a function that associates a non-negative real number $V(\mathcal{B})$ to all subsets of nodes $\mathcal{B}$; this value only depends on the state of nodes inside $\mathcal{B}$, and it is zero if $\mathcal{B}$ is not a clique. The energy function $\mathcal{E}$ is said to derive from the potential V if we can write:

$$\mathcal{E}(\mathbf{s}) = \sum_k \sum_{\mathcal{B} \in C_k} V(\mathcal{B}).$$

When $\mathcal{E}$ derives from V, the *local energy* of node n consists of those terms in $\mathcal{E}(\mathbf{s})$ that involve s_n, i.e.,

$$\mathcal{E}_n(s_n, (s_i)_{i \neq n}) = \sum_k \sum_{\mathcal{B} \in C_k \,:\, n \in \mathcal{B}} V(\mathcal{B}).$$

We define the Gibbs measure associated with an energy function $\mathcal{E}$ and with temperature $T > 0$, as the following probability distribution on the state of the graph:

$$\pi(\mathbf{s}) = e^{-\frac{\mathcal{E}(\mathbf{s})}{T}} / \left(\sum_{\mathbf{s}' \in \mathcal{S}^K} e^{-\frac{\mathcal{E}(\mathbf{s}')}{T}} \right). \quad (2)$$

This distribution has two important properties:

- it favors states of small energy, especially when the temperature T is small;
- it is a Markov random field; conditionally on the states for all neighbors of n, the state of node n is independent of the states of all non-neighbor nodes $k \neq n$.

The *Gibbs sampler* is a procedure where each node n updates its own state according to the following algorithm: (the transitions of all the nodes can occur in an asynchronous way, for instance using an independent exponential timer): given the state of all other nodes than n, node n computes the following probability on $\mathcal{S}$:

$$\mu(s) = \left(e^{-\frac{\mathcal{E}_n(s,(s_i)_{i\neq n})}{T}} \right) \Bigg/ \left(\sum_{s' \in \mathcal{S}} e^{-\frac{\mathcal{E}_n(s',(s_i)_{i\neq n})}{T}} \right), \; s \in \mathcal{S}. \quad (3)$$

Node n then samples a random variable with law μ on $\mathcal{S}$, independently of everything else, and makes a transition to the sampled state. It should be clear from the form of μ that transitions to states of smaller local energy are favored compared to states of higher energy.

When T is fixed, the Gibbs sampler drives the graph to a steady state distributed according to the Gibbs measure (2). Notice that μ only depends on the state of the neighbors of node n. In this sense, the Gibbs sampler is a distributed procedure. The annealed Gibbs sampler combines the above procedure and a slow decrease of T. When T decreases to 0 with time $t > 0$ like $1/\log(1+t)$, we get convergence to a collection of states of minimal global energy. For more on the matter, see e.g. [13].

In the following section, we show how the above procedures can be used to solve the problems of optimal channel selection and optimal user association.

B. Performance Objectives

The objectives for our algorithms are defined as follows.

1) AP channel selection: As noted in Subsection III-B, the higher the interference in the network, (i) the lower the SINR of the users, and (ii) the smaller the time share received by the APs (due to MAC contention). Both factors collectively degrade the users' throughput. Thus, from a heuristic viewpoint, minimizing the total interference in the network should result in improvement of users throughput. Hence our first objective is to find a channel allocation which minimizes the *total interference* received by all APs, namely:

$$\mathcal{F}\left((c_a)_{a \in \mathcal{A}}\right) = \sum_{a \in \mathcal{A}} \left(N_a + \sum_{b \in \mathcal{A} \,:\, b \neq a} s_{\mathrm{CH}}(a,b) P_b(a) \right), \quad (4)$$

where N_a denotes the total thermal noise plus the interference from non-802.11 sources on this channel, and $P_b(a)$ the power of the signal received at a from AP b. The function $\mathcal{F}$ depends on $(c_a)_a$ not only through $s_{\mathrm{CH}}(a,b) = 1_{c_a = c_b}$ but also through N_a and $P_b(a)$ which may both depend on c_a and c_b. Here and in what follows, we do not write the dependence in $(c_a)_a$ to simplify the notation. In practice, we can limit the inner sum to those APs b which are within communication range of AP a, or equivalently assume that $s_{\mathrm{CH}}(a,b)$ is 0 whenever a and b are not within communication range of each other (which we assume to be symmetric).

Let us now see how to cast this in the Gibbsian framework. We take as nodes the APs and say that two APs are neighbors if they are within communication range. The state of AP a is of course its channel c_a. We can expand (4) as:

$$\begin{aligned} \mathcal{F} &= \sum_{a \in \mathcal{A}} N_a + \sum_{\{a,b\} \subseteq \mathcal{A}} s_{\mathrm{CH}}(a,b)(P_b(a) + P_a(b)) \\ \mathcal{F} &= \sum_{\mathcal{B} \subseteq \mathcal{A}} V(\mathcal{B}), \end{aligned}$$

where V denotes the potential function defined, for all subsets $\mathcal{B}$ of $\mathcal{A}$, by

$$\begin{aligned} V(\mathcal{B}) &= N_a && \text{for } \mathcal{B} = \{a\}\ , \\ V(\mathcal{B}) &= s_{\text{CH}}(a,b)(P_a(b)+P_b(a)) && \text{for } \mathcal{B} = \{a,b\}\ , \\ V(\mathcal{B}) &= 0 && \text{for } |\mathcal{B}| \geq 3\ . \end{aligned}$$

Thus $\mathcal{F}$ derives from a potential function. The corresponding local energy of AP a is then given by:

$$\mathcal{F}_a = \sum_{a\in\mathcal{B}} V(\mathcal{B}) = N_a + \sum_{b\neq a} s_{\text{CH}}(a,b)(P_b(a)+P_a(b))\ .$$

Assuming symmetry in power and attenuation, the above equation can be further simplified to

$$\mathcal{F}_a = N_a + \sum_{b\neq a} 2 s_{\text{CH}}(a,b)P_b(a).$$

This expression shows that in this symmetric case, the local energy function can be measured locally by AP a: the term N_a is the ambient noise around AP a, and the term $\sum_{b\neq a} s_{\text{CH}}(a,b)P_b(a)$ is the amount of power a receives from all other APs b within communication range and operating on the same frequency.

When the symmetry assumption cannot be made, APs need to advertise their nominal power (this feature is currently supported by IEEE 802.11k). Note that even in this case, the local energy of AP a only depends on the channels selected by its neighbors, which shows that the advertisements in question remain local.

2) User association: For the user association problem, we assume that each AP has selected some channel. We follow the same Gibbsian methodology. The set of nodes is the set of users. The state of user u is the AP a_u it associates with, so that the state space of a user is the set of APs it can possibly associate with (e.g. the set of APs within its communication range, or the closest APs for each channel and within communication range). Two users are neighbors if there exists an AP with which both could possibly associate. We aim at minimizing the global energy function defined as the sum of the aggregated transmission delays of all users:

$$\mathcal{E}\left((a_u)_{u\in\mathcal{U}}\right) = \sum_{u\in\mathcal{U}} \frac{1}{r_u}\ , \tag{5}$$

where r_u is the long term throughput of user u as given by (1). The aggregated transmission delay of user u should not be confused with the data unit transmission delay $d(u)$ of this user. It may be interpreted as the average delay between the transmission of two data units to this user. The fairness obtained when minimizing the sums of the mean delays of all users of a network, was first introduced under the name of *minimal potential delay fairness* by Massoulié and Roberts [2]. It belongs to the so called α-fair class.

Let us denote by $\{u,v\} \subseteq \mathcal{U}$ a pair of distinct elements of $\mathcal{U}$. Using the results and the notation of §III-B, we see that $\mathcal{E}$ can be simplified as [2]:

$$\begin{aligned} \mathcal{E} &= \sum_{u\in\mathcal{U}} \frac{1}{r_u} = \sum_{u\in\mathcal{U}}\sum_{v\in\mathcal{U}} \frac{s_{\text{AP}}(u,v)d(v)}{M(a_u)} \quad \text{using (1).} \\ &= \sum_{u\in\mathcal{U}} \frac{d(u)}{M(a_u)} + \sum_{u,v\in\mathcal{U}\,:\,u\neq v} \frac{s_{\text{AP}}(u,v)d(v)}{M(a_u)} \\ &= \sum_{u\in\mathcal{U}} \frac{d(u)}{M(a_u)} + \sum_{\{u,v\}\subseteq\mathcal{U}} \frac{s_{\text{AP}}(u,v)d(v)+s_{\text{AP}}(u,v)d(u)}{M(a_u)}. \\ \mathcal{E} &= \sum_{\mathcal{V}\subseteq\mathcal{U}} V(\mathcal{V})\ . \end{aligned}$$

The key observation is that the above cost/energy function also derives from a potential function V defined on the subsets of $\mathcal{U}$ by:

$$\begin{aligned} V(\mathcal{V}) &= d(u)/M(a_u) && \text{for } \mathcal{V} = \{u\} \\ V(\mathcal{V}) &= \frac{s_{\text{AP}}(u,v)d(u)+s_{\text{AP}}(u,v)d(v)}{M(a_u)} && \text{for } \mathcal{V} = \{u,v\} \\ V(\mathcal{V}) &= 0 && \text{for } |\mathcal{V}| \geq 3\ . \end{aligned}$$

Thus, the corresponding local energy of user u is given by:

$$\mathcal{E}_u = \sum_{u\in\mathcal{V}} V(\mathcal{V}) = \frac{1}{M(a_u)}\left((K_{a_u}-1)\cdot d(u) + \sum_{v\in\mathcal{U}_{a_u}} d(v)\right), \tag{6}$$

where K_a denotes the number of users associated with a (including user u). This local energy only depends on the state a_u of user u and that of its neighbors. We assume here that for all a, the AP channel access time $M(a)$ is not a function of the state $(a_u)_u$ of the users, which is reasonable under our saturated downlink traffic assumption as long as as each AP has at least one user.

Notice that this local energy is the sum of two terms. The first term can be seen as the sum of the additional potential delay experienced by other users due to u associating with this AP. This will be called the *social cost*. The second term consists of the sum of the data unit transmission delays of all users affiliated with the AP, and may be seen as the delay experienced by user u because of the other users. We call it the *selfish cost*.

The AP channel access time $M(a) \in [0,1]$, which is the long-term fraction of time for which AP a acquires the wireless channel for its transmissions, can easily be measured (and advertised) by the AP. For a given association to a, the term $d(u)$ can be estimated via the SINR of the signal received from a. If each AP a advertises the total number of its users, K_a will be known to all users that might associate with a. Finally, if each AP advertises its aggregated transmission delay, the sums $\sum_{v\in\mathcal{U}_a} d(v)$ will also be available to all these users. In this sense, the local energy of a given state (or association) can be computed by each user from local information received by APs within communication range.

C. Algorithms

1) AP Channel Selection: We denote the collection of AP channels by $c = (c_a)_{a\in\mathcal{A}} \in \mathcal{C}^{\mathcal{A}}$. In our solution, each AP maintains an exponential timer, with mean t_a (an exponential timer is a timer which expires as per a random variable that

[2] As before, we do not explicitly write the state variables $(a_u)_u$ as arguments of $\mathcal{E}$ to keep notation light. Similarly, to be exhaustive, one should write $d(v, a_v)$ in place of $d(v)$ in the following expressions for $\mathcal{E}$ and V.

is exponentially distributed); whenever this timer expires, the AP performs the following transition:

Algorithm 1 *(Annealed Gibbs sampler for AP transition)*

1) *Compute the temperature parameter:* $T = \frac{T_0}{\log_2(2+t)}$.
2) *For all channels c, compute the local energy experienced by the AP a on this channel :* $\mathcal{F}_a(c) = N_a + 2\sum_{b\in\mathcal{A}\,;\,c_b=c} P_b(a)$.
3) *For all channels c, compute the probability*

$$\pi(c) = \left(e^{-\frac{\mathcal{F}_a(c)}{T}}\right) \Bigg/ \left(\sum_{c\in\mathcal{C}} e^{-\frac{\mathcal{F}_a(c)}{T}}\right) \ .$$

4) *Sample a random variable with law* π *and choose a channel according to this random variable.*

In this algorithm, t is an *age variable* (that roughly represents the time elapsed since initialization of the network) and T_0 is the parameter that determines the speed of convergence.

The version of this algorithm with a temperature kept constant to T is the plain Gibbs sampler for AP transition and temperature T. It is referred as Algorithm 2.

A greedy algorithm : Alg. 1 is probabilistic in nature, and it requires that APs maintain a loose synchronization through a common time t. In practice, an AP could also follow the simpler greedy local optimization, whenever its timer expires:

Algorithm 3 *(Greedy AP transition)*

$$\textit{Choose channel } c = \operatorname{argmin}_{c\in\mathcal{C}}(\mathcal{F}_a(c)).$$

Alg. 3 may be seen as a limit case of Alg. 1 when the temperature goes to zero: state updates are not randomized but always chosen to minimize the local energy. Instead of a logarithmic temperature decrease, which drives in Alg. 1 the distribution towards minimal energy states, Alg. 3 minimizes the local energy observed for each transition[3]. In other words, Alg. 3 can be interpreted as an aggressive version of Alg. 1.

2) User Association: The algorithm for user association is very similar to the one proposed in the previous section. In this algorithm, the user collects information pertaining to the number of users and the aggregated transmission delay for every AP a it can possibly associate with. The user then computes the local energy $\mathcal{E}_u$, given by (6), that it would experience if it would associate with a.

Each user maintains an age variable t and an exponential timer with mean t_u. Whenever its timer expires, the user performs this transition:

Algorithm 4 *(Annealed Gibbs Sampler for User Transition)*

Follow the same steps as in Alg. 1 (starting with Temperature T_0'*) and choose to associate with AP* a *with probability* π *defined from the local energy* $\mathcal{E}_u(a)$ *as defined in (6).*

The plain Gibbs sampler, that is the version of this algorithm for a constant temperature, is referred to as Algorithm 5. The greedy version of Alg. 4 may also be defined as follows.

[3]Experimental observations reveal that the auto-channel selection algorithm in Cisco APs uses a similar greedy heuristic to avoid congested channels, but details could not be confirmed due to proprietary issues.

Algorithm 6 *(Greedy User transition)*

$$\textit{Choose AP } a = \operatorname{argmin}_{a\in\mathcal{A}}\left(\mathcal{E}_u(a)\right) \ .$$

D. Analysis for Static Population

We characterize in this section the convergence of the algorithms we presented above. We start with a result on the plain Gibbs samplers (Alg. 2 and 5).

Theorem 1 *For a fixed population of APs and users that implement Alg. 2 (resp. 5) with temperature* T*, the distribution of the state of the network converges in variation to the Gibbs distribution (4) associated with total interference (resp. (5) associated with total potential delay) for temperature* T*.*

Both algorithms can be described by a finite state homogeneous Markov chain. A classical reversibility argument shows that the Gibbs distribution is the invariant probability measure for these chains. By arguments similar to Example 6.5 p.288 in [13], one gets that the convergence has a geometric speed.

Theorem 2 *For a fixed population of APs and users that implement Alg. 1 and 3, there exist values of* T_0, T_0' *such that AP channel selection and user association converge to a state of minimum energy as time goes to infinity.*

$$\mathcal{F}\left((c_a)_{a\in\mathcal{A}}\right) \to \min_{(c_a)_{a\in\mathcal{A}}} \mathcal{F}$$

$$\textit{and } \ \mathcal{E}\left((a_u)_{u\in\mathcal{U}}\right) \to \min_{(a_u)_{u\in\mathcal{U}}} \mathcal{E} \ .$$

Proof: The proof is analogous to that of Example 8.8 p.311 in [13]. The network evolves according to a *strongly ergodic* non-homogeneous Markov chain: it converges in variation to a limit distribution that only puts positive probability mass on the states of minimum global energy. ∎

Theorem 3 *For a fixed population of APs and users that implement Alg. 3 and 6, the resulting AP channel selection and user association verify, as time goes to infinity*

$$(c_a)_{a\in\mathcal{A}} \to (\tilde{c}_a)_{a\in\mathcal{A}} \ \textit{ and } \ (a_u)_{u\in\mathcal{U}} \to (\tilde{a}_u)_{u\in\mathcal{U}}$$

where $\tilde{c}$ *and* $\tilde{a}$ *are local minima, in the following sense:*

$$\textit{For any } a \in \mathcal{A},\ \ \tilde{c}_a = \operatorname{argmin}_{c\in\mathcal{C}}\mathcal{F}_a(c)$$

$$\textit{For any } u \in \mathcal{U},\ \ \tilde{a}_u = \operatorname{argmin}_{a\in\mathcal{A}}\mathcal{E}_u(a) \ .$$

Proof: From the definition of Alg. 3, the global energy function $\mathcal{F}$ can only decrease after each transition, since the transition of a given AP leads to a decrease in the sum of energies of all subsets containing that AP, while leaving the energy of other subsets unchanged. As the state space of AP channels is finite, this sequence converges after a finite number of steps. Similarly, the sequence of user associations also converges in a finite number of steps. ∎

The annealed sampler is the only algorithm which converges to a collection of states of minimum energy, for all fixed topologies. This comes at a cost since the convergence requires a slow (logarithmic) cooling scheme.

In contrast, the plain sampler, which may use any fixed temperature T, does not converge to a minimum energy state, but only to a random state distributed according to the Gibbs distribution. This distribution gives higher probability to small energy states, and its speed of convergence is geometric (in the number of transitions). Choosing to implement the plain sampler is therefore trading the quality of the optimum against the speed of convergence.

The difference between the annealed sampler and the greedy algorithm is that the latter can get blocked in a local minimum of the energy. Choosing to implement the greedy algorithm is therefore trading long-term efficiency for quick improvement and simpler implementation. It is not difficult to find situations where a local minimum exists. Here is a simple example pertaining to channel selection: 8 APs are arranged on a linear grid where neighboring nodes are at distance 1; we assume that the power received from another AP at distance 2 is 1, and that the power received from another AP at distance 1 is 3. For two-non overlapping channels, 0 and 1, the channel configuration 01100110 is a local minimum (i.e. no node has an incentive to change its channel state). The energy (i.e. the sum of total power received) is 18 for this configuration. However, the minimum energy, which is attained for configuration 01010101, is 12. Hence, the greedy algorithm cannot escape from this local minimum and one needs Alg. 1 to reach a global minimum.

Extensive simulations have shown that for large random populations of APs/users, the local minima that are found by the greedy algorithms provide excellent approximations to the results obtained by the annealed samplers (see Section V).

E. Dynamic Population

The annealed Gibbs sampler can be adapted to a slowly varying dynamic topology. To deal with the case of a dynamic population of APs, each AP should maintain the list of APs that are in its neighborhood, and their respective signal strengths. If this list changes (because of an AP joining or leaving), the age variable of the AP is re-initialized to zero. Notice that this requires the propagation of the global variable t throughout the network. For highly dynamic topologies, a reasonable and practical solution, studied below, is to use the plain samplers. For cases where the population of users varies (it may vary quite rapidly compared to the set of the active APs), reasonable options would be to use either the Gibbs sampler with fixed temperature or Alg. 6.

V. Simulation Results

In this section, we use simulations to evaluate the convergence and scalability properties, as well as the performance benefits of the proposed algorithms. We use a customized simulator for large network topologies, and the OPNET simulator for small network topologies. Our customized simulator (which we call the Gibbs simulator), uses a precomputed mapping between SINR and data rate (obtained from measurements on our testbed - see Section VI). It incorporates (i) the max-min fair throughput sharing of 802.11 for downlink traffic within the same cell, (ii) the path-loss and shadowing effects, and (iii) the random arrival and departure of APs and users. To remain scalable, the Gibbs simulator does not incorporate packet level effects and, hence, approximates contention at the MAC layer using the following simple rule. The AP channel access time is equal to the inverse of the number of APs in its contention domain, i.e. if the AP contends with 2 more APs, they each access the channel for 1/3 of the time. OPNET, on the other hand, has an accurate model for simulating 802.11 MAC dynamics at the packet level, but does not scale well[4].

A. Validation of Simulation Methodology

Our simulator does not incorporate MAC layer contention. Consequently, it may underestimate the impact of interference when multiple APs are in the same contention domain. To understand the implication of this limitation we performed the following simulations. For 2000 randomly and independently generated network topologies with average cell radius of 50m and 11 orthogonal channels, we ran Alg. 3. We observed that 99.8% APs did not have any other AP in their contention domain. For the remaining 0.2% APs, there was just one more contending co-channel AP. This shows that for an average cell radius of 50m (or larger) and 11 orthogonal channels, MAC contention between co-channel APs is almost completely eliminated when Alg. 3 is used for channel allocation. Thus, selecting channels using Alg. 3 sufficiently resolves MAC contention, thereby rendering our Gibbs simulator appropriate for further exploration.

We first study the convergence properties of the proposed algorithms using the Gibbs simulator in Subsection V-B. To study the throughput benefits of Alg. 1-3, we compare the performance with a random channel allocation scheme. Note that although Gibbs channel selection results in contention-free channel allocation, random channel allocation may result in channel assignment in which a large fraction of APs may contend with each other. For this setting, it is necessary to take the 802.11 MAC dynamics into account. We therefore use OPNET to compare the channel selection schemes, and these results are presented in Subsection V-C. Finally, the simulation results for Alg. 4-6 are presented in Subsection V-D.

B. Convergence and Stability

In the customized flow level simulator, we simulate network topologies where APs and wireless users are located in a square of size 2000m x 2000m. The path loss exponent is set to 3.0 and all nodes use a default transmit power of 20dBm. The SINR to data rate relationship is derived from real measurements using our testbed from §VI. For all the simulations presented, an AP assesses the need for a transition on average every 3 hours (t_a in Alg. 1), while users test the need for a transition on average every 15 minutes (t_u in Alg. 4). For ease of comparison, the same overall mean numbers are used in all cases: 500 APs and 5000 users, corresponding to an average cell radius of 50m, and about 10 users per cell. More precisely, we consider APs and users distributed according to one of the following distributions:

- **Homogeneous topology**: the locations of APs and wireless users are sampled according to independent Poisson point processes (PPP) in the square.

[4]On a Pentium III PC a single simulation run of 10 minutes for a topology of 30 WLAN nodes in OPNET takes around 10 hours.

- **Sporadic topology**: same as above but now users are sampled according to a non-homogeneous PPP that results in regions with very high density of users. We configured 10% of the APs on the plane to have 10 times higher user intensity than the global intensity.

We consider both (i) a static topology, and (ii) a dynamic topology, where users and APs may join and leave the network across time. To facilitate the comparison between the static and dynamic topologies, APs and users join and leave such that the number of APs and the number of users remains unchanged. In order to demonstrate the benefits of the proposed algorithms, we compare their performance with what could be considered the *current best practice*: the case where APs select their channels randomly and users affiliate with the AP with the strongest signal strength.

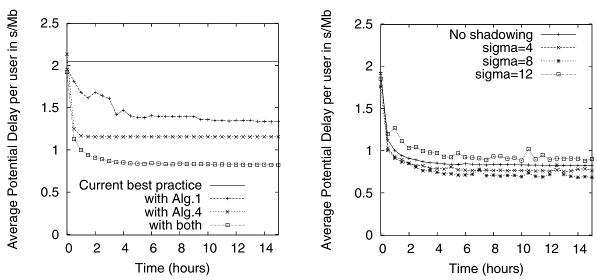

Fig. 1. Average Potential Delay per user seen over time for a static topology: with different algorithms (left), impact of shadowing (right).

1) Static Sporadic Topology: We first look into static sporadic topologies, where at time zero, the APs and users choose their channels and associations according to the current best practice. Fig. 1 (left) presents the evolution of the sum of potential delay (i.e. the value of the energy $\mathcal{E}$ over time) and the improvement observed through the introduction of the proposed algorithms. From Fig. 1 (left), we see that the mean potential delay improves when Alg. 1 is used. This improvement is reached on average after two transitions, corresponding in our setting to six hours. We have obtained similar qualitative results on static homogeneous topologies.

The Gibbsian algorithm for user association (Alg. 4) shows similar convergence patterns. Referring to Fig. 1 (left), the improvement is even more pronounced as the minimal potential delay stabilizes after one hour, which corresponds to four user transitions on average. Finally, the use of both algorithms combined offers the best performance. Most of the gain is obtained within one hour. The greedy versions of Alg. 1 and Alg. 4, i.e., Alg. 3 and Alg. 6 respectively, show identical performance improvements, but the corresponding results have been omitted due to space constraints.

2) Impact of Shadowing: We simulate the same topologies as above in the presence of shadowing with a shadowing standard deviation, σ, varying from 0 to 12dB. The results are presented in Fig. 1 (right). We see that even when σ is large, our algorithms still converge, and perform quite well. The improvement decreases as σ increases. This is because as σ becomes large, locality in the optimization problem is lost, as the shadowing component in the overall channel gain becomes more dominant than the fixed path loss component. Nevertheless, our algorithms are robust in this context.

3) Dynamic Topology: In this sub-section we simulate dynamic changes to the topology where users can join and/or leave the network. The join and leave events have the following effect: when a user joins he/she automatically selects an AP to affiliate with. A user leaving has no other effect than improving the throughput of the other users in the same cell. When an AP joins, it triggers an immediate transition to channel selection, but users may not immediately associate with it. When an AP leaves, all its users immediately trigger a transition in order to choose another AP to associate with.

We have introduced exponential timers that alter the topology every 90s in mean. The time between topological changes for the user population is hence on average ten times smaller than the time between two transitions for the same user. Each of these topological changes may alter up to 10% of the population of users, so that the proportion of the population that changes between two transitions, that we call *turnover*, may reach 100%.

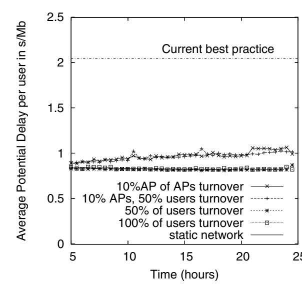

Fig. 2. Average potential delay as a function of time for dynamic topologies.

The results are shown in Fig. 2. The most remarkable observation is that dynamic user population appears not to be an issue in terms of convergence and algorithmic efficiency, since a turnover up to 100% is still almost indistinguishable from the static case.

C. Frequency selection

In this section, we demonstrate the benefits of our frequency selection algorithm (Alg. 3, chosen for simplicity), for a small topology (20 APs and 20 users) by incorporating the effects of the 802.11 MAC. For this, we use OPNET. All the nodes use a default transmit power of 20dBm, a carrier sensing threshold of -90dB, and a path loss exponent of 3.0. The APs are randomly and uniformly distributed in the network with an average inter-AP distance of about 100m. We use 802.11a PHY layer that supports 11 orthogonal channels, so that there are approximately twice as many APs as channels. Note that OPNET does not incorporate rate adaptation. Hence we allocate one user per AP, and the user location is chosen close to its respective AP (within 10m) to ensure that all the APs and the users can sustain the maximum transmission rate of 54 Mbps. This enables us to study the impact of channel selection in isolation. Saturated UDP traffic, comprising 1500 Byte packets, is sent over the downlink.

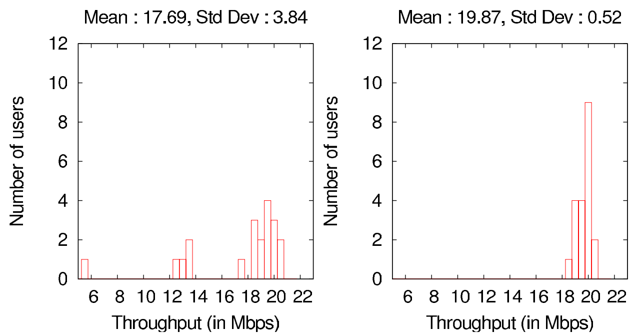

Fig. 3. Histograms of the long-term user throughput for the 40 node topology: random channel assignment (left); channel assignment using Alg. 3 (right).

Referring to Fig. 3 (left), we see that with the random channel assignment scheme, the average user throughput is 17.69Mbps. Five out of 20 users receive a throughput less than 14Mbps, while one user receives a throughput less than 6Mbps. This is because random channel assignment results in two or more co-channel APs in the same contention domain, which need to time-share wireless access. On the other hand, the channel assignment performed by Alg. 3 ensures that there are no co-channel APs in the same contention domain. This is reflected in the fact that all the users get the same maximum throughput of about 20Mbps (see Fig. 3 (right)). We also note that Alg. 3 results in about 12.3% improvement in the average throughput. The Jain Fairness Index for the considered topology for Alg. 3 is 0.999, while for the random channel assignment scheme is 0.957. Thus, the above results demonstrate the effectiveness of the proposed channel selection algorithm in mitigating interference, and thereby resulting in improved throughput.

D. User association

The throughput performance of Alg. 4 is studied using the Gibbs simulator. The results for Alg. 5 and 6 are similar and we do not show them due to space constraint. Referring to Fig. 1 (left), we see that the average potential delay decreases from 1.45 second per Mb to 0.8 second per Mb when Alg. 4 is used in addition to Alg. 1. In Fig. 4, we plot the empirical distribution of the throughput obtained by the entire user population across all 2000 topologies in log-log scale. On the left, we show the distribution achieved by the current best practice. We notice that when users choose APs with the strongest signal, the throughput distribution obtained has high variance. It spans from maximal values obtained when a single user is associated with an AP (21Mb/s), to small throughput values obtained by a non-negligible fraction of the user population (43 users on average have less than 200kb/s, 1400 users on average have less than 500kb/s).

In contrast, our combined algorithms, shown in Fig. 4 (right), lead to a more even distribution (all users, except from 7, receive more than 200kb/s; all except 125 receive more than 500kb/s). With our algorithms, there are fewer users with very high throughput which is primarily due to the load balancing feature of our algorithms. However this is achieved at the cost of a slightly lower average throughput. Similar performance has been observed with static homogeneous topologies.

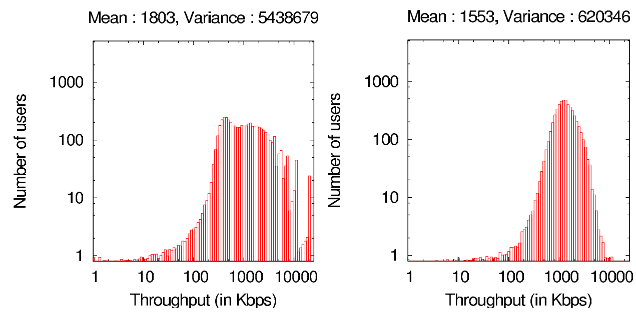

Fig. 4. Histograms of the distribution of long-term throughput, in the case of a sporadic topology: closest association (left); association using Alg. 4 (right). AP channel selection have been made according to Alg. 1.

We have also studied the performance of Alg. 4 for the scenarios of incremental deployment and selfish users. The latter is similar to the user association algorithm proposed in [8]. However, we do not present the results here due to space constraints. Please see [10] for details.

VI. Experimentation on a Testbed

We prototyped the proposed algorithms on a small scale testbed. The prototyping served two purposes: (i) to show that the algorithms can indeed be implemented on today's hardware, and (ii) show that for simple topologies, the user association algorithm, (we have chosen Alg. 6 for simplicity), results in optimal bandwidth sharing. However, the channel selection algorithms require a large number of APs in the testbed (at least twice the number of orthogonal channels), and although we implemented Alg. 3, we were not able to evaluate its performance extensively due to the small size of our testbed. We are currently addressing the challenges that crop up when experimenting with the proposed algorithms over a large testbed.

A. Implementation Details

In our algorithms, APs and users need to evaluate, whenever a timer expires, whether a transition to another channel or another AP is needed. The main requirements are the following: (i) APs should be able to measure the total interference on each channel. (ii) Each user should be able to evaluate the SINR received from all the APs within range, the number of users of each AP, and each AP's associated aggregated transmission delay (see Eq. (6)). (iii) Each AP should be able to notify all its users about an upcoming change in its operating channel.

Of the above, (i) is currently supported by 802.11 hardware. Functions (ii) and (iii) are being addressed within the 802.11k and 802.11h task groups of the IEEE. IEEE 802.11k defines a framework to facilitate radio resource management within which WLAN devices exchange statistics, say to make more informed roaming decisions. The number of users supported by an AP is part of the 802.11k specification and the aggregated transmission delay and AP channel access time could easily be added and communicated to the hosts inside the Beacon probes. IEEE 802.11h defines the mechanisms that need to be implemented by an AP for Dynamic Frequency Selection (DFS) and Transmit Power Control (TPC). Within the proposed 802.11h standard, APs can initiate channel switch announcements to their users, i.e, function (iii).

Since many of the above functionalities are not available on the current 802.11 hardware, we implemented some of them on the Intel 2915ABG cards used in our testbed nodes. Following is a list of modifications that were made to the firmware of the cards. (i) APs can scan all frequencies and measure the total amount of interference on each of them. They can select the operating frequency that minimizes the cost function $\mathcal{F}$. (ii) APs can measure $M(a)$ and $\sum_{v\in\mathcal{U}_a} d(v)$. Both values are computed over 5 second intervals and their exponential weighted moving average, with a 0.8 weight, are advertised through the AP beacon frames once every 30 seconds to avoid firmware instability. (iii) Users can decode the new metrics in the beacon frames and select an AP for association so as to minimize $\mathcal{E}$.

B. Network Topology and Experimental Results

The testbed consists of 2 Dell Inspiron desktops as APs, and 4 IBM T30 laptops as users. Both platforms use the Intel 2915ABG wireless cards with the open source `ipw2200` client driver that has been modified to support the AP functionality discussed above. The topology used for evaluating the performance of the Gibbsian algorithms is as follows. The test machines are deployed in a corridor in such a way that one user is placed on either side of each AP, with 2 users between the 2 APs. Using Alg. 3, the APs pick two non-overlapping channels for their operation. We then initialize the 4 users in turn starting from the outside users and moving toward the area between the 2 APs. After each user affiliates with the AP of its choice, a 1 Mb/s CBR source starts from the AP to the user. We show the results of our experiment in Fig. 5. The two bottom figures show the throughput achieved by each user, along with the time when they become active and which AP they select. The upper figure plots the aggregated transmission delay for each AP. We notice that the aggregated transmission delay exhibits increases upon the affiliation of new users tracking the activity of the AP. More importantly, we clearly see that the fourth user decides to affiliate with AP2 given its smaller aggregated transmission delay compared to AP1 (the user receives similar RSSI from both APs). Thus, through this simple network topology, we can see how the user association algorithm performs load balancing for optimal bandwidth sharing.

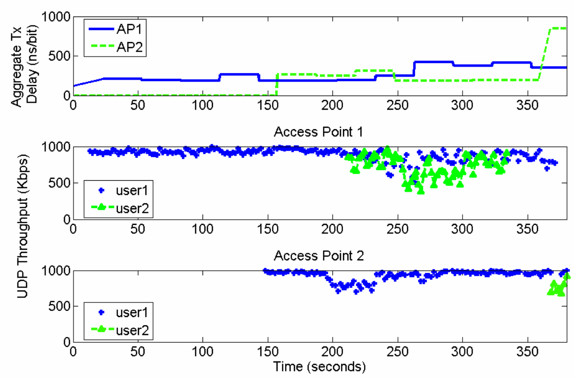

Fig. 5. Aggregate transmission delay and throughput for the 6-node testbed.

VII. Conclusion

Using the Gibbs sampler framework, we designed fully distributed algorithms for (i) channel selection for interference mitigation, and (ii) user association for fair and optimal sharing of bandwidth between users. We demonstrated that distributed decisions which take into account the individual gain as well as the social cost can lead to efficient spectrum usage, and improved performance. This was proved in the context of the minimal potential delay fairness. The fact that this notion of fairness is amenable to a distributed optimization within the 802.11 context is one of the key scientific observations of the paper.

Implementation of the proposed algorithms relies on firmware modifications, and requires features that could be incorporated within the efforts of the IEEE 802.11 task groups. The actual implementation was shown to be feasible with today's hardware on a small scale testbed. Experimentation over a larger testbed and extension of the analytical framework to study mesh networks is part of our future work.

Acknowledgments

We would like to thank Alexandre Proutière for pointing to us the minimal bandwidth delay fairness. We would also like to thank Changwen Liu, York Liu, Jing Zhu, and Xingang Guo from Intel Corporation, Portland, USA for facilitating our performance evaluation, based on OPNET and modifications to the Intel wireless card firmware and driver.

References

[1] A. Akella, G. Judd, P. Steenkiste, and S. Seshan, "Self management in chaotic wireless deployments," in *ACM MobiCom*, 2005.

[2] L. Massoulié and J. Roberts, "Bandwidth sharing: objectives and algorithms," *IEEE/ACM Trans. Netw.*, vol. 10, no. 3, pp. 320–328, 2002.

[3] B.-J. Leung, K.K.; Kim, "Frequency assignment for IEEE 802.11 wireless networks," in *IEEE VTC*, vol. 3, 2003, pp. 1422–1426.

[4] Y. Bejerano, S. Han, and L. Li, "Fairness and load balancing in wireless LANs using association control," in *ACM MobiCom*, 2004.

[5] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *ACM MobiCom*, 2003.

[6] A. Mishra, D. Agrawal, V. Shrivastava, S. Banerjee, and S. Ganguly, "Distributed channel management in uncoordinated wireless environments," in *ACM MobiCom*, 2006.

[7] A. Kumar and V. Kumar, "Optimal association of stations and APs in an IEEE 802.11 WLAN," in *NCC*, 2005.

[8] T. Korakis, O. Ercetin, S. Krishnamurthy, L. Tassiulas, and S. Tripathi, "Link quality based association mechanism in IEEE 802.11h compliant wireless LANs," in *1st Workshop RAWNET*, 2005.

[9] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh, "A client-driven approach for channel management in wireless LANs," in *IEEE Infocom*, 2006.

[10] B. Kauffmann, F. Baccelli, A. Chaintreau, K. Papagiannaki, and C. Diot, "Self organization of interfering 802.11 wireless access networks," INRIA, Tech. Rep., 2005.

[11] K. Sundaresan and K. Papagiannaki, "The need for cross-layer information in access point selection algorithms," in *ACM IMC*, 2006.

[12] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda., "Performance anomaly of 802.11b," in *IEEE Infocom*, 2003.

[13] P. Brémaud, *Markov Chains, Gibbs Field, Monte Carlo Simulation and Queues*. Springer-Verlag, 1999.